BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

REMARKS

The application has been amended and is believed to be in condition for allowance.

Claims 1-26 are pending, claims 25-26 being newly presented.

There are no formal matters pending.

Claims 1-24 stand rejected as anticipated by WALDO et al. 6,708,171.

WALDO et al. has been offered as disclosing a request source client with a stub search means for sending a stub request formed from a stub name and client identifier to a server. WALDO et al. column 17, lines 35-39 was offered:

> **Upon receipt of the request from the client computer 11(n), the control 28 in the server 12(m) searches the lookup service 400 for the stub 404 corresponding to the requested service (step 604). If there are no matches found, the control 28 returns a null value (steps 606 and 608).**

This passage fairly discloses that the server searches a lookup service for a stub corresponding to the requested service. The requested service corresponds to the recited "stub name". However, there is no disclosure as to using the recited client identifier to search for a stub

corresponding to the requested service in an appropriate runtime environment.

As to the WALDO et al. lookup service 400, see Figure 4. The lookup service 400, located on a server 12(m), maintains a collection of "service items" 410-418 each of which represent an instance of a service available. Each service item 410 contains a service ID 402 that uniquely identifies the service item, a stub 404 providing code that programs use to access the service, and a collection of attributes 406 that describe the service. This appears to be a lookup table listing stubs available on the server.

Applicant, however, sees no disclosure concerning plural runtime environments or methodology concerning determining the appropriate runtime environment from the submitted client identifier.

The Official Action also points to column 9, beginning at line 29:

> **[the client computer] will attempt to obtain the stub class instance 30 from … the server computer 12(m) …, and enable the stub class instance 30 to be dynamically loaded in the execution environment 20 for the invoking class instance 22. A reference to the remote object may be received, for example, either as a return value of another remote method invocation or as a parameter that is received during another remote method invocation. The**

stub class instance may be dynamically loaded into the execution environment in a manner similar to that used to load class instances 22 in the execution environment 22.

...

However, if the stub class loader 33 is not otherwise notified of which server computer 12(m) maintains the class which implements a method which may be invoked remotely, it may use the nameserver computer 13 to provide that identification.

The above is general discussion of stub searching and adds nothing concerning plural runtime environments or methodology concerning determining the appropriate runtime environment from the submitted client identifier.

To make the recitations concerning the server support of plural runtime environments, claim 1 has been amended to recite explicitly that the server supports plural different runtime environments and returns to the request source client the stub selected for the runtime environment of the request source client. It is also explicitly recited that the stub search interface (of the server) utilizes the client identifier to determine the appropriate runtime environment for the request source client from the plurality of different runtime environments supported by the server.

WALDO et al. is not found to teach or suggest these features. For each of the above reasons, WALDO et al. is not believed to be anticipatory.

As to claim 2, there is recited stub sets prepared for each type of the different client runtime environments. To clarify, the form (but not the substances) of claim 2 has been amended. As amended, claim 2 reads that the server further comprises a plurality of stub sets, each of the stub sets corresponding to one of the plural runtime environments and including stubs to be used together with a skeleton used in the server at the time of remote method invocation from the request source client. See application Figure 1, e.g., stub search interface 27 connected to storage section 29 illustrating stub class sets for three java runtime environments.

From specification page 19:

"Referring back to Fig. 1, upon receiving a stub request with a designated stub class name ..., **the client identification section 28 identifies a client identifier sent together and determines which set in the stub classes 26-1 to 26-3 should be defined as a search target.** The client identification section 28 holds, e.g., **a table 28a that holds the correlation between client identifiers and the names of sets of stub classes** (the names of the stub classes 26-1 to 26-3) and specifies a corresponding set of stub classes by

searching the table 28a using the client identifier. The stub search interface 27 searches for the stub class having the stub class name designated by the stub request from one of the stub classes 26-1 to 26-3, which is determined by the client identification section 28, and returns the stub class to the client computer 1, 2, or 3 of the request source."

Also see on specification page 2:

"the client identifier can be any information with which **a runtime environment** where a downloaded stub is executed can be directly or indirectly specified.

"The stub search interface 27 in the server computer 4 determines, on the basis of the client identifier received by the client identification section 28, which set in the stub classes 26-1 to 26-3 should be defined as a search target. The stub class having the stub class name designated by the stub request is searched for from the determined set of stub classes 26-1, 26-2, or 26-3 (step S10)."

As discussed in Summary of the Invention section on specification pages 10-11, the present invention provides a stub search loading system which allows a plurality of client computers **using different types of Java runtime environments** to download, from **a single server computer**, a stub class that can be used for remote method invocation in each client computer.

In order to achieve this, there is provided a stub search loading system for downloading a necessary stub in the appropriate runtime environment by request source client sending a stub request formed from a stub name and client identifier, and the server returning to the request source client the stub appropriate for a runtime environment of the request source client **on the basis of the designated stub name and client identifier.**

WALDO et al. do not make disclosures in this regard.

Applicant does not see any disclosure comparable to Figure 1 stub search interface 27 with client identification section 28 (table 28a) linked to storage section 29 comprising plural stub classe sets, each stub class set being dedicated to one of plural runtime environments.

In amended claim 5, the recitations are sending a stub request formed from a stub name and client runtime identifier from the request source client to the server. This clarifies the role of the previously recited client identifier consistent with the disclosed invention. Claim 5 continues to recite that upon receiving the stub request, that the stub returned to the requesting source client is selected on the basis of the designated stub name and the client runtime identifier so that the stub is appropriate for a runtime environment of the requesting source client.

WALDO et al. do not make this disclosure.

Claim 6 has been amended to clarify that the server, on the basis of the designated client runtime identifier, selects one stub set from the plural stub sets, where each stub set has stubs prepared for one of the different runtime environments.

WALDO et al. do not make this disclosure.

The other original claims have been similarly amended.

New claims 25-26 include recitations of the invention requiring i) a stub search section for installation on plural client computers of differing runtime environments; and ii) a stub search interface for installation on a server computer in communication with the plural client computers.

Claim 25 specifies that the stub search section permits each client computer to act as a requesting client computer and to issue a stub class request to the stub search interface. Thereafter, the requesting client computer may download from the server computer the requested stub class for remote method invocation to the server computer, the downloaded stub class being provided from the server computer to the requesting client computer in the runtime environment of the requesting client computer.

Claim 25 requires that the stub class request comprise a stub class name and a client identifier, the runtime environment of the requesting client computer being

determinable from the client identifier, where the stub search interface responds to the stub class request, on the basis of the designated stub class name and client identifier, by returning to the requesting client computer the requested stub class appropriate for the runtime environment of the requesting client computer. WALDO et al. do not make this disclosure.

Claim 26 is believed patentable in its own right as WALDO et al. is not found to disclose a stub search interface responding to the stub class request by using the client identifier to determine which of a plurality of stub class sets for the differing runtime environments is appropriate for searching, i.e., of the correct runtime environment.

Properly considering the recitations of the pending claim set, applicant believes it is clear that WALDO et al. neither teaches nor suggests the presently recited invention. Therefore, reconsideration and allowance of all the pending claims are respectfully requested.

Please charge the fee of $86 for the extra independent claim and $18 for the dependent claim added herewith, to Deposit Account No. 25-0120.

The Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 25-0120 for any

determinable from the client identifier, where the stub search interface responds to the stub class request, on the basis of the designated stub class name and client identifier, by returning to the requesting client computer the requested stub class appropriate for the runtime environment of the requesting client computer. WALDO et al. do not make this disclosure.

Claim 26 is believed patentable in its own right as WALDO et al. is not found to disclose a stub search interface responding to the stub class request by using the client identifier to determine which of a plurality of stub class sets for the differing runtime environments is appropriate for searching, i.e., of the correct runtime environment.

Properly considering the recitations of the pending claim set, applicant believes it is clear that WALDO et al. neither teaches nor suggests the presently recited invention. Therefore, reconsideration and allowance of all the pending claims are respectfully requested.

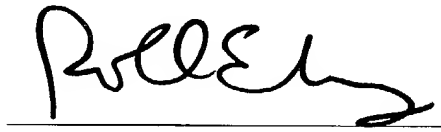
Please charge the fee of $86 for the extra independent claim and $18 for the dependent claim added herewith, to Deposit Account No. 25-0120.

The Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 25-0120 for any

additional fees required under 37 C.F.R. § 1.16 or under 37 C.F.R. § 1.17.

Respectfully submitted,

YOUNG & THOMPSON



Roland E. Long, Jr., Reg. No. 41,949
745 South 23rd Street
Arlington, VA 22202
Telephone (703) 521-2297
Telefax (703) 685-0573
(703) 979-4709

REL/lk